JTRS JPO SCA Extension Workshop NAVSYS Experience with GPP & FPGA Interfacing April 29th-30th, 2004

Frank Carpenter
NAVSYS Corporation
14960 Woodcarver Road
Colorado Springs, CO 80921
Phone: 719.481.4877 email:
frankc@navsys.com
www.navsys.com

NAVSYS SGR GPP/FPGA Interface

- The interface between the GPP and FPGAs for our Software GPS Receiver (SGR) product is defined in a portable fashion using VHDL files.
- A simplified VHDL parser in the GPP scans for register definitions and maps addresses in the processor to registers in the FPGA.
- Both Programmable I/O (PIO) and Direct Memory Access (DMA) are supported.
- Reduces human errors and maintains consistency by using same VHDL interface definition in generating FPGA bitfiles as is used with the application.
- The same VHDL interface definition can be used when multiple FPGAs are loaded with the same bitfile. The register address mapping is offset by the FPGA base address.
- Within a VHDL interface definition file multiple devices can be defined and mapped to a single FPGA. An extensible array of devices can be created with multiple FPGAs. e.g. 12 GPS correlation engines spread across 3 FPGAs.

SGR VHDL Register Definition

- VHDL Control and Data Register examples from interface .vhd file

```
CtrlRegN.Timer_X                                <= RegSelN(1);

DataRegN.LdCaCodeDelPhase_DumpCaEarlyI(0) <= RegSelN(2);
```

- VHDL Parser Output

```
CAC DMA Buffer Offset 0x00000200 Read Buffer @ 0x81A2E008
Write Buffer @ 0x81A2D008 Priority 255

CAC0.Timer.CW Buffer Offset 0x000001 PIO Offset
0x101004 Type Ctrl Write

CAC0.CaCodeDelPhase.DW Buffer Offset 0x000002 PIO Offset
0x100008 Type Data Write
```

Future Directions

- Need ability to translate between Verilog, VHDL, high-level source (C++/C), CORBA, IDL, XML, etc.
- Intermediary syntax / language like IDL or XML.
- Dynamic run-time configuration or static compile-time configuration.
- Need ability to reload FPGAs or DSPs on the fly.
- Dynamic provides ability to tailor FPGA code to an application without changing hardware or software.
- Dynamic supports ability to load different FPGA device types.
- Static can provide faster startup but at cost of pairing application/FPGA loads.

Conceptual IDL for GPP/FPGA Interfacing

```
module FpgaDevice {
  enum FpgaDataType {
    Control, Data
  };

  typedef ULong FpgaOffset;
  typedef ULong RegisterSelect;

  typedef struct FpgaDataDefinition {
    FpgaDataType     Type;
    FpgaOffset       Offset;
    RegisterSelect   RegSel;
  };

  /* Manifest Constant Control/Data Register Definitions */
  FpgaDataDefinition Timer_X { Type = Control; Offset = 0x04; RegSel =
   1;};

  FpgaDataDefinition LdCaCodeDelPhase_DumpCaEarlyI { Type = Data; Offset =
   0x08; RegSel = 2;};
};
```